Method for protecting a character entered at a graphical interface

Field of the Invention

This invention relates to the field of computer security and more particularly to a method and system for protecting a character entered at a graphical interface.

Background of the Invention

Password is commonly required for enabling access across a network to an application hosted by a service provider. In a web-centric environment, a user of the service is required to enter his password into a textbox in the browser, which is then submitted to the server application using SSL for authentication. Unfortunately, this does not protect the password sufficiently as the client computer is vulnerable to security breach.

Monitoring software present in the client computer can be recording key-presses, mouse-clicks, and screenshots without the user's knowledge. This means that a hacker who has access to the monitoring software can steal the user's password, regardless of whether the password is entered using the keyboard or by clicking on a graphical keypad on the screen.

Publicly accessible computers, like those found in airports or internet cafes, are especially vulnerable to such hacking as users have neither control nor knowledge over what are installed on the computers. It is important, especially for the service provider, to secure client computers to prevent such hacking activity. In addition, other confidential user information, like user ID or account number, are also vulnerable.

Presently, the best technique to thwart such hacking activity is to use scanning software to scan for monitoring software and to detect key and mouse logging activities. The disadvantage of this technique is that the scanning software needs to be installed on the client computer. This may not always be possible as the service provider cannot dictate what is installed on the user's computer, or the user may be using a public terminal and has no permission to install anything. Another disadvantage is that the scanning software may need regular updating to function properly which can be a costly process. Hence, it can be seen that this technique is not a satisfactory solution.

The problem is therefore how to obtain password, or other confidential information, in such a way that is safe from the prying "eyes" of monitoring software.

Summary of the Invention

The invention described herein permits the user to decipher a keypad image while monitoring software can capture merely unreadable portions of the keypad image. In furtherance of this purpose, imaging techniques, including data partitioning and random distribution, are combined with the known capability of the human vision system to fuse dissimilar images into a single image.

Accordingly, the present invention provides a method of protecting a character entered at a graphical interface. The inventive method comprises the steps of: generating a set of images that form a complete image of a keypad having a button-to-character assignment; displaying said graphical keypad using said image set; and, obtaining the character of a selected button using said button-to-character assignment. The inventive method can be repeated with a different button-to-character assignment in each repetition to obtain a sequence of characters.

In consequence of the inventive method, the entered character is protected against monitoring software as monitoring software can capture merely unreadable portions of the complete keypad image.

In furtherance of this purpose, the present invention also provides a method of generating a set of images from a complete image of a character belonging to a character set. The inventive method comprises the steps of: computing the visible probabilities of all possibly illuminated pixels in a complete image; partitioning said pixels into groups based upon visible probability; and, distributing the illuminated pixels in said complete image among two or more images based upon pixel group.

The step of displaying the graphical keypad can comprise the step of displaying the images in the image set sequentially and cyclically at a fast refresh rate. In accordance with the displaying step, a strobed display of the complete keypad image is viewed by the user.

In an embodiment, the entered character sequence could represent confidential information, for example, password, account number, or user ID, and therefore the present invention could be implemented to protect entry of confidential information against monitoring software.

Brief description of the drawings

There are presently shown in the drawings embodiments which are presently preferred, it being understood, however, that the invention is not limited to the precise arrangements shown.

FIG. 1 is a flowchart illustrating the inventive method for protecting a character sequence entered at a graphical interface implemented in a client-server network environment.

FIG. 2 illustrates an exemplary graphical keypad seen by the user.

FIG. 3 illustrates an exemplary keypad with a different button-to-character assignment from that in FIG. 2.

FIG. 4 illustrates an exemplary set of three images that could be sent to the client application for display.

FIG. 5 illustrates the formation of the character "4" using visual persistence.

FIG. 6 illustrates the formation of the character "4" with a different pixel distribution from that in FIG. 5.

FIG. 7 is an exemplary pattern that illustrates a typical character display mechanism used in commercially available seven-segment LED display.

FIG. 8 is a flowchart illustrating the inventive method for pixel distribution.

FIG. 9 illustrates the pixel groups generated for the seven-segment character display mechanism.

Description of the preferred embodiments

The preferred embodiment is described in terms of an implementation of the inventive method in two network-connected applications, wherein the user uses a client application to enter a character sequence to a server application, although the invention could be implemented similarly in other environment, for example, in a single application or in several network-connected applications.

FIG. 1 is a flowchart illustrating the inventive method for protecting a character sequence entered at a graphical interface in a client-server network environment. The method begins at step 200 after communication has been established between the client and server applications. In step 200, the server application initializes the entered character sequence to an empty string. It then creates, in step 202, a new button-to-character assignment. In step 204, it creates a set of images that form a complete image of a keypad having the button-to-character assignment and sends the newly created image set to the client application for display. After sending the images, the server application waits for the client application.

FIG. 2 illustrates an exemplary complete image of a keypad having an exemplary button-to-character assignment. In the preferred embodiment, the character sequence comprises only numeric characters from "0" to "9". However, the present invention is not limited in this regard and could be implemented for any character sequence, alphanumeric or otherwise. In FIG. 2, the ten buttons are assigned to unique characters; the top left-hand corner button is assigned to character "1", the button next to its right is assigned to character "4", the next right button is "5", and so forth. The server application uses a random algorithm to randomly assign each button to a character in the preferred embodiment, although the present invention could be implemented using other assignment algorithm to perform step 202.

In step 100, the client application uses the received image set to display a graphical keypad to the user by employing the visual persistence of the human vision system to fuse dissimilar images into a single image. FIG. 4 illustrates an exemplary received image set comprising portions of the exemplary complete keypad image illustrated in FIG. 2. The client application sequentially and cyclically displays the three images in FIG. 4 at a fast refresh rate on its screen. By visual persistence, the cyclic pattern of images is integrated into a strobed display of the graphical keypad illustrated in FIG. 2. This integration is illustrated in FIG. 5 for the formation of the character "4" button in FIG. 2. The three images M1, M2 and M3 in FIG. 5 are obtained from the corresponding three images in FIG. 4.

In accordance with the present invention, the client application has no knowledge of the button-to-character assignment; it just displays whatever it receives. This reduces the security risk of exposing confidential information in the client application. The client screen does not display a complete keypad image at any one instance. Consequently, monitoring software can capture merely unreadable portions of the complete keypad image. Hence, when the user selects a keypad button, the entered character is safeguarded.

Specifically, a set of guidelines exist for displaying the images on the client screen while minimizing perceived flicker and eyestrain. The client application synchronizes each image drawing with the client monitor's vertical retrace period to prevent image tearing. The background and foreground colors of the keypad buttons are blue and red respectively in the preferred embodiment, although the present invention could be implemented in other colors. The number of images in an image set is three in the preferred embodiment, although the present invention could be implemented with other number of images. The number of images in an image set can be dependent on the monitor's refresh rate and the human vision system's integration time.

After displaying the graphical keypad, the client application waits for the user to select either a keypad button or a submit button in step 102. The user enters his first character by selecting a keypad button, using a mouse or other input means. The client application detects the keypad button selection and, in step 104, removes the graphical keypad from the client screen and informs the server application about the selected button by sending screen coordinates or other identification means. In step 206, the server application obtains the entered character from the selected button using the button-to-character assignment and adds the selected character to the entered character sequence. The server application then abandons the existing button-to-character assignment. This ends the first cycle of steps performed to get the first character of the character sequence. Subsequently, for each remaining character in the character sequence, the cycle starts from step 202, wherein a new button-to-character assignment is created for the next character entry by the user.

In the inventive method, the button-to-character assignment changes randomly every time; this is illustrated in FIG. 3 which shows another exemplary keypad with a different button-to-character assignment from that in FIG. 2. Consequently, faithful reproduction of mouse clicks captured by monitoring software is useless.

The cycle ends when the user clicks on a submit button in step 102. The client application then, in step 106, informs the server application that no more characters will be entered, removes the displayed keypad, and terminates. In step 208, the server application forwards the entered character sequence to a separate module for processing and terminates.

In step 204 of FIG. 1, the server application generates a set of images that form a complete keypad image. Generally, the step of generating such image set can comprise the step of partitioning a complete keypad image into portions showing partial keypad information. The present invention does not limit the algorithm that could be used for generating such image set. Specifically, the present invention provides a method (hereinafter "inventive method for pixel distribution") for generating a set of images from a complete image of a character belonging to a character set.

The inventive method for pixel distribution is described in terms of the implementation of the preferred embodiment, wherein a seven-segment character display mechanism is employed, although the invention could be implemented similarly in other display mechanisms, for example, bit-mapped font rendering.

As illustrated in FIG. 2 and FIG. 5, the preferred embodiment uses a seven-segment character display mechanism, illustrated in FIG. 7, to display the ten characters in the keypad. In the seven-segment character display mechanism, some of the seven segments in FIG. 7, from S1 to S7, are illuminated on a screen to display a character. It can be seen that the seven segments effectively comprise all the possibly illuminated pixels in a complete image of a character.

The flowchart in FIG. 8 illustrates the inventive method for pixel distribution. The method starts by partitioning the possibly illuminated pixels of the employed character display mechanism into pixel groups. In step 300, the visible probability of each possibly illuminated pixel is computed. The visibility probability of a possibly illuminated pixel is the likelihood that the possibly illuminated pixel is illuminated when a character is displayed. In step 302, the number of pixel groups is defined as the closest integer that is greater than or equal to the decimal quotient obtained from dividing the number of possibly illuminated pixels by the number of images in an image set. The size of each pixel group is initialized to the number of images in an image set. If the sum of the size of all the pixel groups is greater than the number of possibly illuminated pixels, then a number of pixel groups equal to the difference are reduced in size by one. If not, the pixel groups retain their initial size. The possibly illuminated pixels are now assigned to the empty pixel groups. In step 304, each pixel group, in descending order of size, is given the possibly illuminated pixel with the next highest visible probability until the group is full.

In the preferred embodiment, according to steps 300, 302 and 304, the three pixel groups, as illustrated in FIG. 9, are obtained. The labels S1 to S7 refer to the seven possibly illuminated pixels in the employed display mechanism illustrated in FIG. 7, and the numbers in brackets are the visible probabilities of each possibly illuminated pixel.

After the pixel groups for the employed display mechanism are generated, the method distributes the illuminated pixels in a given complete image of a character among an image set of empty images in step 306. The method randomly places illuminated pixels into images while ensuring each illuminated pixel in an image comes from a different pixel group.

FIG. 5 illustrates an exemplary illuminated pixel distribution for the character "4". The illuminated pixels in the complete image of this character are S2, S3, S6 and S7 according to FIG. 7. Based upon the pixel groups in FIG. 9, S2 and S3 are from the same group and, therefore, appear in two different images M3 and M2 respectively. In accordance with the inventive method for pixel distribution, the pixel distribution randomly changes each time step 306 is executed. This is illustrated in FIG. 6 which shows a different exemplary pixel distribution for the same character "4".

In consequence of the inventive method for pixel distribution, it is difficult to determine the character represented by an image set through examining one of the images since the pixel distribution for the character changes randomly and infrequently-used possibly illuminated pixels are not found in the same image.

In general, the inventive method for pixel distribution is used for displaying character securely. In the preferred embodiment, the inventive method for pixel distribution is used on the characters in a keypad to produce a set of images that form a complete image of the keypad.